AMENDMENTS TO THE CLAIMS

Please amend the claims as indicated below. The language being added is underlined ("__") and the language being deleted contains either a strikethrough ("——") or is enclosed by double brackets ("[[ ]]").

1. (Currently Amended) A communications processor for implementing scheduling processes and reducing the processing effort for determining a minimum value of a plurality of values stored in source registers and determining an index value of a source register having the minimum value, the communications processor comprising:

a destination register;

a first source register storing a first value, wherein the first source register comprises S bits, and wherein the first value comprises N lower bits of the first source register;

a second source register storing a second value, wherein the second source register comprises S bits, and wherein the second value comprises N lower bits of the second source register;

means for comparing the first value stored in the first source register with the second value stored in the second source register;

means for storing the first value in the destination register when the first value is less than or equal to the second value; and

means for concatenating the index value with the second value into a concatenated value and storing the concatenated value in the destination register when the second value is less than the first value, wherein the index value is stored in an

upper (S-N) bits of the concatenated value and the second value stored in the N lower bits of the concatenated value, wherein the first source register and the second source register each include an active status bit to indicate a status of the respective register, and wherein a value of a register having an active status is less than a value of a register having an inactive status.

2. (Original) The processor of claim 1, wherein the means for comparing, the means for storing and the means for concatenating are adapted to execute sequentially within one processor cycle.

3. (Original) The processor of claim 1, wherein the first source register and the destination register comprise a same register.

4. (Original) The processor of claim 1, wherein the second source register and the destination register comprise a same register.

5. (Original) The processor of claim 1, wherein the first value is stored in N low-order bits of the first source register and the second value is stored in N low-order bits of the second register, N being an integer value.

6-12. (Canceled)

13. (Currently Amended) A method implemented as instructions for manipulating a network processor for implementing scheduling processes and reducing a number of network processor cycles for determining a minimum value and a corresponding index value of a plurality of source registers of [[a]] the network processor, the method comprising the steps of:

~~assigning~~ initializing a destination register with an index value and a value of a first source register from among the plurality of source registers;

for each of the plurality of source registers, comparing a value stored in the source register with a value stored in a destination register;

concatenating the value stored in the source register with an index value associated with the source register and storing the concatenated value in the destination register when the value stored in the source register is less than the value stored in the destination register; and

setting active status bits in the source register and in the destination register such that a value of a register having an active status is less than a value of a register having an inactive status, wherein comparing, concatenating, and storing are implemented by a single processor instruction and performed within a single processor cycle by the network processor, wherein each of the plurality of values represents a due timestamp of a corresponding input queue.

14-15. (Canceled)

16. (Currently Amended) The method of claim 13, wherein ~~each of~~ the plurality of values ~~represents~~ representing a due timestamp of a corresponding input queue [[for]] is used for implementing Weighted Fair Queuing.

17. (Currently Amended) A customer premise equipment (CPE) comprising:

a network interface operably connected to a first network segment;

a second network interface operably connected to a second network segment; and

a processor operably connected to the network interfaces and being adapted to:

compare a first value stored in a first source register of the processor with a second value stored in a second source register of the processor;

store the first value in a first destination register of the processor when the first value is less than or equal to the second value; and

store the second value in the first destination register of the processor and an index value in a second destination register of the processor when the second value is less than the first value, the index value representing the second source register, wherein the first source register and the second source register both include a status identifier for indicating whether the respective register is active or inactive, wherein the status identifier comprises the least significant bit of the value in each respective register.

18. (Original) The CPE of claim 17, wherein the processor is further adapted to compare the first and second values and store values in the destination registers within one processor cycle.

19. (Original) The CPE of claim 17, wherein the first value and second value each represents a due timestamp of a corresponding input queue for implementing Weighted Fair Queuing for at least one data stream between the first network segment and the second network segment.

20. (Currently Amended) A communications processor for determining a maximum value of a plurality of values stored in source registers and determining an index value of a source register having the maximum value for scheduling transmission of data, the communications processor comprising:

a destination register;

a first source register storing a first value;

a second source register storing a second value;

means for comparing the first value stored in the first source register with the second value stored in the second source register, wherein the first source register and the second source register both include a status bit to indicate whether the respective register is active or inactive, and wherein a value of a register having an active status is greater than a value of a register having an inactive status;

means for storing the first value in the destination register when the first value is greater than or equal to the second value; and

means for concatenating the index value with the second value into a concatenated value and storing the concatenated value in the destination register when the second value is greater than the first value.

21. (Original) The processor of claim 20, wherein the means for comparing, the means for storing and the means for concatenating are adapted to execute sequentially within one processor cycle.

22. (Original) The processor of claim 20, wherein the first source register and the destination register comprise a same register.

23. (Original) The processor of claim 20, wherein the second source register and the destination register comprise a same register.

24. (Original) The processor of claim 20, wherein the first value is stored in N low-order bits of the first source register and the second value is stored in N low-order bits of the second register, N being an integer value.

25 - 31. (Canceled)

32. (Currently Amended) A method implemented as instructions executed by a communications processor for implementing scheduling processes and[[for]] reducing a number of processor cycles for determining a maximum value and a corresponding index value of a plurality of source registers of [[a]] the processor, the method comprising the steps of:

~~assigning~~ initializing a destination register with an index value and a value of a first source register from among the plurality of source registers;

for each of the plurality of source registers,

comparing a value stored in the source register with a value stored in a destination register, wherein values stored in the plurality of source registers represent due timestamps of corresponding input queues used for transmission of data packets;

concatenating the value stored in the source register with an index value associated with the source register and storing the concatenated value in the destination register when the value stored in the source register is greater than the value stored in the destination register; and

setting active status bits in the source register and in the destination register such that a value of a register having an active status is greater than a value of a register having an inactive status, wherein comparing, concatenating, and storing are performed within a single processor cycle.

33. (Original) The method of claim 32, wherein the steps of comparing, concatenating, and storing are implemented by a single processor instruction.

34. (Canceled)

35. (Currently Amended) The method of claim 32, wherein ~~each of~~ the plurality of values ~~represents a due timestamp of a corresponding input queue~~ representing due timestamps are used for implementing Weighted Fair Queuing.

36. (Currently Amended) A customer premise equipment (CPE) comprising:

a network interface operably connected to a first network segment;

a network interface operably connected to a second network segment; and

a processor operably connected to the network interfaces and being adapted to:

compare a first value stored in a first source register of the processor with a second value stored in a second source register of the processor;

store the first value in a first destination register of the processor when the first value is greater than or equal to the second value; and

store the second value in the first destination register of the processor and an index value in a second destination register of the processor when the second value is greater than the first value, the index value representing the second source register, wherein the first source register and the second source register both include a status identifier for indicating whether the respective register is active or inactive, and wherein the processor is configured to transmit data based on the status identifier.

37. (Original) The CPE of claim 36, wherein the processor is further adapted to compare the first and second values and store values in the destination registers within one processor cycle.

38. (Original) The CPE of claim 36, wherein the first value and second value each represents a due timestamp of a corresponding input queue for implementing Weighted Fair Queuing for at least one data stream between the first network segment and the second network segment.